Journal of Computing and Information Technology - CIT 14, 2006, 4, 337–342
doi:10.2498/cit.2006.04.10

Combined Model for Congestion Control

Nguyen Hong Van[1], Oliver Popov[2], Iskra Popova[2]

[1]DSV, Stockholm University and KTH, Kista, Sweden
[2]ITM, Mid Sweden University, Sundsvall, Sweden

The growth of multimedia applications on the Internet made at least one fifth of the total network traffic to run over UDP. Unlike TCP, UDP is unresponsive to network congestion. This may cause, inter alia, bandwidth starvation of responsive flows, severe and prolonged congestions or, in the worst-case scenario, a congestion collapse. Hence, the coexistence of both protocols on fair-share premises converges towards impossibility. The paper deals with a new approach to solving the problem of taming down the unresponsive flows. By using some of the desirable properties of mobile agents, the system is able to control the influx of non-TCP or unresponsive flows into the network. Various functions performed by mobile agents monitor non-TCP flows, calculate sending rates and modify their intensity according to the needs of the network to attain as good performance as it is possible.

Keywords: TCP flows, non-TCP flows, congestion control, mobile agents, simulation.

1. Introduction

The proliferation of networked multimedia parallels the growth of Internet. Despite novel techniques for data compression and multicast for data transmission, multimedia applications are bandwidth intensive, delay sensitive, and somewhat loss tolerant. TCP being both a reliable and fair protocol (retransmits every lost or corrupted packet and slows down in case of congestion) is mostly suited for file transfers, terminal work and web browsing. This usually does not work in transporting interactive video and sound, where reliability is a weakness rather than a strength, and consequently UDP is the protocol of choice.

UDP has no mechanisms either to detect, or to control congestion, which classifies it as an unresponsive protocol. When low capacity link becomes a bottleneck and the network may enter into a state of congestion, UDP maintains its transmission rate. It may use almost all capacity of that link. while the self-clocking TCP, as congestion responsive, will slow down and thus decrease the goodput that can eventually go to zero. The phenomenon is known as a congestion collapse since most of the network resources transmit undelivered packets [1].

When the number of TCP flows in the Internet is prevalent, the stability of the network is guaranteed by the congestion control mechanisms as an integral part of the transport protocol. In the presence of UDP, the situation radically changes, which makes any co-existence of different transport protocols a virtual impossibility and the appearance of congestion a reality.

Recent research has focused on studying and resolving this problem, as in Network Border Patrol, [2], where all data flows are monitored and their sending rates are accordingly adjusted via traffic shapers placed at the edge routers. Another solution is the Datagram Congestion Control Protocol (DCCP) [3], a sort of a blend between UDP and TCP, where the complexity of the latter is reduced just to its congestion control features. The suggested solutions are still being studied and experimented with, which makes the question of a suitable congestion control strategy when socially responsible and socially irresponsible protocols have to work together, as it is the case today on the Internet.

A novel model named Combined Model for Congestion Control (CM4CC) is in the centerpiece of the article. The principal goal of the model is (1) to prevent the network from congestion collapse, and (2) to make the network recover in a fast manner (enter a normal mode of operation after a congestion event).

2. Combined Model for Congestion Control (CM4CC)

One of the characteristics of mobile agents is the ability to reduce network traffic and have acceptable level of performance in unreliable and low bandwidth networks, [4]. In CM4CC, mobile agents are used to monitor non-TCP flows, collect information about their sending and receiving rate, as well as manage these flows. The agents calculate loss and the maximum allowed sending rates, and control them indirectly so they do not exceed the values corresponding to the current network state.

This is a host-centric approach, i.e. the end hosts control and regulate non-TCP flows before they enter the network. These should reduce the number of dropped packets at the routers, which results in lower loss rates and higher throughput.

There are no requirements as to the modifications and changes of the network devices. A single condition is the existence of an operating environment for the mobile agents. There is no need for exchange of messages over the network in regular time intervals. A rather small amount of network resources is employed to move mobile agents and to collect and relay the information. Furthermore, the agents are created on as-needed basis that translates into a low overhead at the end-hosts. The possible complexity induced by the mobile agent paradigm is reduced by the exploration of the built-in TCP congestion control.

In Figure 1, the CM4CC shows two different categories of flows, namely TCP or congestion responsive, and non-TCP flows that use mobile agents (marked in gray). CM4CC considers that the network is in congested state when a timeout occurs in at least one TCP flow. Any other state is considered normal. Mobile agents are used for (1) congestion control when the network is in congested state, and (2) for congestion avoidance when it is in normal state.

In total, there are seven agents classified in three groups: management, monitor and control group. Three of them are always on the network, while the rest are created when they are needed.

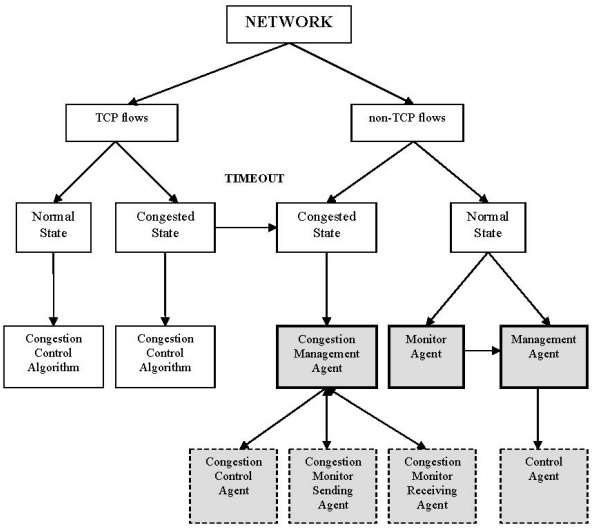

Fig. 1. Conceptualization of CM4CC.

The agents always present are the Management Agent, Congestion Management Agent and the Monitor Agent (they are denoted by bold and solid line boxes). The other four are the Congestion Monitor Sending Agent, the Congestion Monitor Receiving Agent, the Control Agent and the Congestion Control Agent (denoted by dotted line boxes in Figure 1.).

3. Mobile Agents in Action

The network is assumed to be in a normal state when no TCP flow experience timeout. In this case, as shown on Figure 2, mobile agents are used as the congestion avoidance mechanism for non-TCP flows.

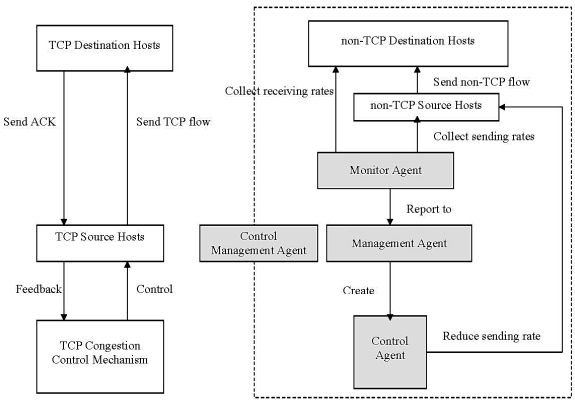

Fig. 2. The system in a normal state.

The network is considered to be congested when the timer in at least one TCP flow has expired, which indicates an imminent slow-start phase for the flow. This prompts the mobile agents to work as a congestion control mechanism for the non-TCP flows.

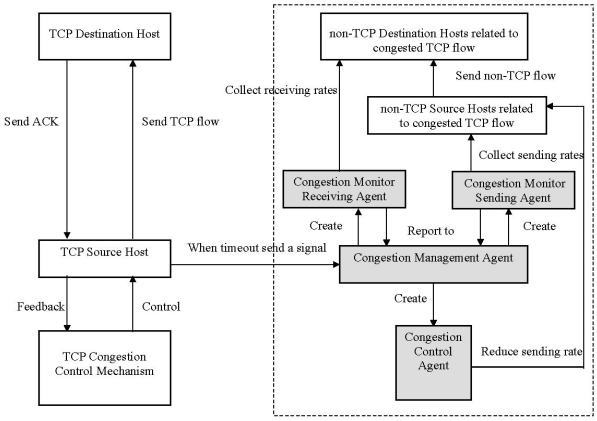

Fig. 3. The system in a congested state.

A non-TCP flow is related to a congested TCP flow when both of them have the same edge router. The Congestion Management Agent does nothing until it receives a signal indicating congestion. When there is at least one non-TCP flow related to a congested TCP flow, the Congestion Management Agent creates the Congestion Monitor Sending Agent and the Congestion Monitor Receiving Agent. These two agents, along with the Monitor Agent, create the group of monitoring agents. They have the responsibility to detect active unresponsive flows and collect information about sending and receiving rates of non-TCP flows. The Monitor Agent has responsibility for all non-TCP flows in the network, while the Congestion Monitor Sending Agent and the Congestion Monitor Receiving Agent monitor only the non-TCP flows related to the congested TCP flows.

The Management Agent and the Congestion Management Agent belong to the group of management agents and are used to coordinate the activities of all mobile agents, which include the policies for unresponsive flows, loss and maximum allowed sending rates, and the creation of control agents when necessary. The Management Agent manages all the hosts in the network. On the contrary, the Congestion Management Agent only manages source and destination hosts, viz. sending and receiving non-TCP flows related to congested TCP flows.

The Control Agent and the Congestion Control Agent are control agents in the model. Their responsibility is to (1) move to the source hosts in which the sending rates of non-TCP flows exceed the maximum allowed sending rates; (2) bring with them the information on the maximum allowed sending rates; (3) adjust and control indirectly sending rate of non-TCP flows through the traffic shaper placed at these source hosts.

4. The Interplay of Sending Rates

Congestion avoidance in a normal state of the network and congestion control in a congested one are attained in CM4CC by reducing the sending rates for the non-TCP flows in both states. The case for the congested state of the network is evident. Regulating sending rates in a normal state reduces the probability of the network entering a congested one. This section states the rules for congestion avoidance and control by CM4CC and makes an estimate of the maximum allowed sending rate that serves as a threshold for the activation of control management agents.

Let us say f_{ij}^k is a non-TCP flow k sent from host i to host j, $i = 1 \ldots n_1$, $j = 1 \ldots n_2$, where $k = 1 \ldots n_3$ with n_1, n_2, n_3 being positive integers. The sending rate and receiving rate for the flow are $SR(f_{ij}^k)$ and $RR(f_{ij}^k)$ respectively. Then, the loss rate for this flow, $LR(f_{ij}^k)$ is determined as:

$$LR(f_{ij}^k) = \frac{SR(f_{ij}^k) - RR(f_{ij}^k)}{SR(f_{ij}^k)}.$$

Let α be the number that represents the error tolerance of transmission media. The values for α can be different, depending on the transmission media used. When the loss rate of any non-TCP flow is less than α, we will assume that this is due to the properties of the media and will consider as if there is no loss at all. In general, multimedia applications are loss tolerant. However, after the loss rate exceeds certain limit, the performance starts to degrade. This

limit is termed as the allowed loss rate for the non-TCP flow, $ALR(f_{ij}^k)$.

The CM4CC model uses two rules for congestion control and congestion avoidance These are:

1. When the network is in a congested state and $LR(f_{ij}^k) > \alpha$, the sending rate of all non-TCP flows related to congested TCP flows is reduced below the Maximum Allowed Sending Rate (MASR), determined as:

$$MASR(f_{ij}^k) = SR(f_{ij}^k) \cdot \left(1 - LR(f_{ij}^k)\right).$$

2. When the network is in a normal state and $LR(f_{ij}^k) > ALR(f_{ij}^k)$, the sending rate of all non-TCP flows is reduced below the Maximum Allowed Sending Rate (MASR), determined as:

$$MASR(f_{ij}^k) = SR(f_{ij}^k) \cdot \cdot \left[1 - \left(LR(f_{ij}^k) - ALR(f_{ij}^k)\right)\right].$$

5. Traffic Controller

Whenever one of the rules stated above holds, the control mobile agents in CM4CC move to the source hosts generating non-TCP flows, where they control the sending rates of the flows. This is done via a tool termed as a traffic controller, which is located in the interface between the host and the network.

Figure 4 shows the traffic controller architecture. It consists of four components: packet filter, flow classifier, rate controller, and per-flow traffic shaper (token bucket). The packet filter filters out the packets from non-TCP flows whose sending rates exceed the maximum allowed sending rate. The function of the flow classifier is to classify these packets into separate flows. The rate controller adjusts the parameters of the traffic shaper based on the feedback information about the maximum allowed sending rate. The per-flow traffic shaper limits the rates of the flows before they actually enter the network.

The Control Agent or the Congestion Control Agent must also deliver the information about each of the controlled non-TCP flows to the

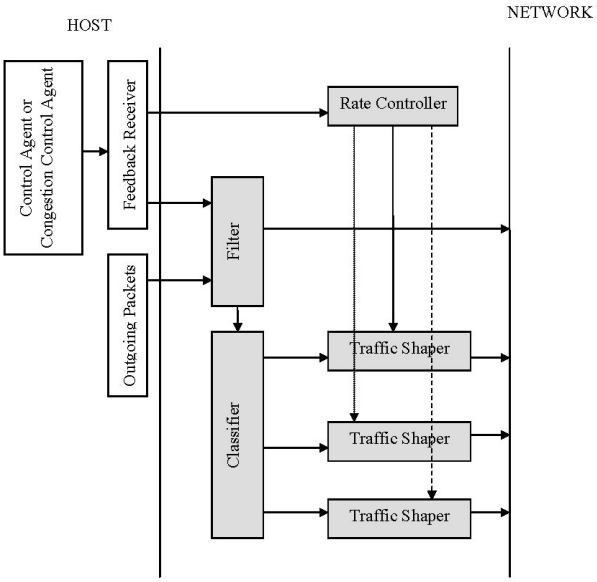

Fig. 4. The architecture of traffic controller.

feedback receiver. The information consists of two parameters: the maximum allowed sending rate and the identity of the flow, e.g. the flow ID in IP v6 or the address of the source host, the destination host, the source port, and the destination port.

The feedback receiver passes the parameters to the packet filter and the rate controller. Outgoing packets whose ID matches the ID parameter are passed to the flow classifier. These are actually the packets belonging to non-TCP flows that need to reduce their sending rate. The flow classifier classifies the packets according to their sending rates. An individual traffic shaper regulates the sending rate for each flow. The traffic shaper parameters are updated and adjusted by the control agent through the rate controller. The packets whose IDs do not match the ID parameter are passed directly to the network.

The parameters in the feedback receiver are valid for a limited time. In the absence of any parameters or their invalidity there are no constraints on the infusion of the flows into the network (or no effect of the sending rates).

The traffic controller, which is placed between a host and the network, does not require substantial changes in the hardware of the host or a modification in the router architecture.

The sending rate of the non-TCP flows is re-

duced at each source host. Consequently, the amount of controlled flows is not very large. For now, the model does not deal with flow classification and maintenance of the state of flows in a network. This can be a demanding problem in itself, especially in large networks with a significant number of flows.

6. Conclusion

Congestion is a serious problem in heterogeneous networks. This is even truer today when the variety of data types present in the network traffic has increased and generates both responsive and unresponsive flows.

The CM4CC regulates the sending rate of both responsive and unresponsive flows by using a combination of classical network management and mobile agent paradigm. The first one deals with responsive flows, while the second one is primarily focused on unresponsive flows.

The approach that uses mobile agents can be considered a reasonable choice for congestion control due to the flexibility and on-the-fly adaptability of the mobile agents. They can be created whenever they are needed to monitor, collect necessary information and control indirectly the sending rates of non-TCP flows. The actions cover both the congested state and the normal state of the network. It should be noted that there are some serious issues concerning security whenever the mobile agent paradigm is invoked. When one takes into account the mobility, the need for penetration across different layers and information extraction and dissemination, the concerns are legitimate. For now, the security issues have been addressed by the parsimonious usage of the mobile agents, both in time and in space. While this minimalistic approach is pursued, other solutions to rectify the problems with security shall be explored too.

Congestion can be prevented or stopped by simultaneously reducing the sending rates of relevant unresponsive flows. Since most of the mobile agents terminate after the completion of their tasks, the implementation and the operation of the model are neither too complex nor too expensive in terms of cost. This also implies the possibility of an improved quality of service in a best-effort network.

We are confident that the proposed model provides a good basis for a unified approach to the solution of the global network congestion. The preliminary results, based on using some simulation tools, such as AgentSpase2, analytical simulations, and the strong and sound theoretical basis, more than justify the pursuit of further research in this area.

References

[1] S. FLOYD, K. FALL, Promoting the Use of End-to-End Congestion Control in the Internet, *IEEE/ACM Transaction on Networking*, Vol. 7, No. 4, August 1999.

[2] C. ALBUQUERQUE, B. VICKERS, AND T. SUDA, Network Border Patrol, *Proceedings of IEEE INFOCOM*, Tel-Aviv, Israel, March 2000.

[3] E. KOHLER, M. HANDLEY, S. FLOYD, Designing DCCP: Congestion Control without Reliability, *Tech. Rep. ICIR*, 2004.

[4] NGUYEN HONG VAN, Mobile Agent Paradigm in Computer Networks, *Proceedings of 6th CARNet User Conference*, Zagreb, Croatia, September 2004.

[5] NGUYEN HONG VAN, O. POPOV, A Conceptual Model for Congestion Management of Internet Traffic Flows, in *Proceedings of CCCT2004*, Austin, Texas USA, Vol. 5, pp. 402–406.

Received: June, 2006
Accepted: September, 2006

Contact addresses:

Nguyen Hong Van
DSV, Stockholm University and KTH
Forum 100
SE-164 40 Kista
Sweden
e-mail: si-hvan@dsv.su.se

Oliver Popov
ITM, Mid Sweden University
Holmgatan 10
SE-851 70, Sundsvall
Sweden
e-mail: oliver.popov@miun.se

Iskra Popova
ITM, Mid Sweden University
Holmgatan 10
SE-851 70, Sundsvall
Sweden
e-mail: iskra.popova@miun.se

NGUYEN HONG VAN is a Ph.D. student at the Department of Computer and Systems Sciences, Stockholm University and Royal Institute of Technology. She works for the Ministry of Science and Technology of Vietnam, and her academic stay in Sweden is made possible through a scholarship from the Swedish government, which is administered by SIDA.

OLIVER B. POPOV is a professor of computer science in the Department for Information Technology and Media at Mid Sweden University, and an associate professor in the Computer and Systems Sciences Department at Stockholm University in Sweden. His research inclinations converge towards computer networks and the Internet technology, ICT4D, research and education networking, and information society.

ISKRA POPOVA is an associate professor of computer science at in the Department for Information Technology and Media at Mid Sweden University. She works is in the areas of routing algorithms, ICT4D, and e-learning.